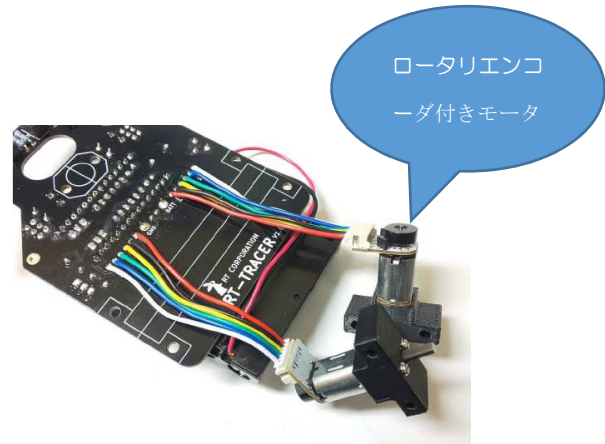
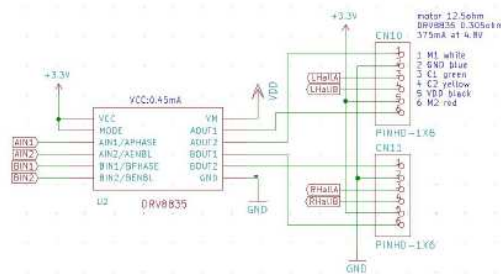


目標 コースを記憶、第2走行しよう!!

1.ロータリエンコーダを理解する。



3.3 モータユニット

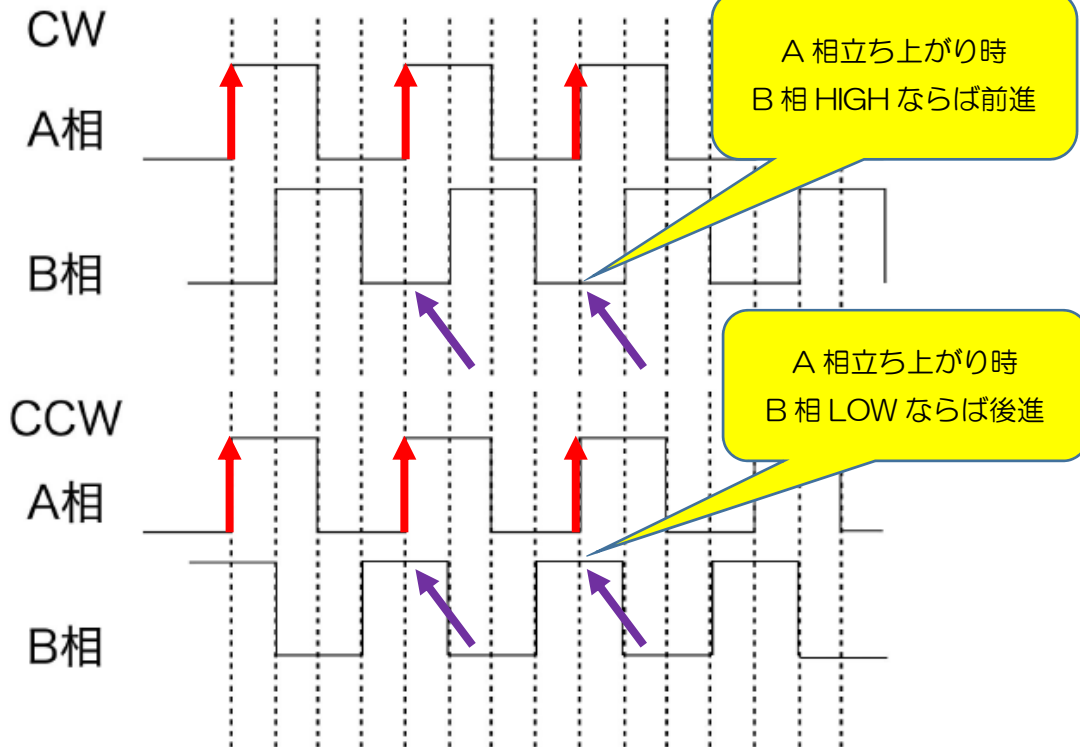


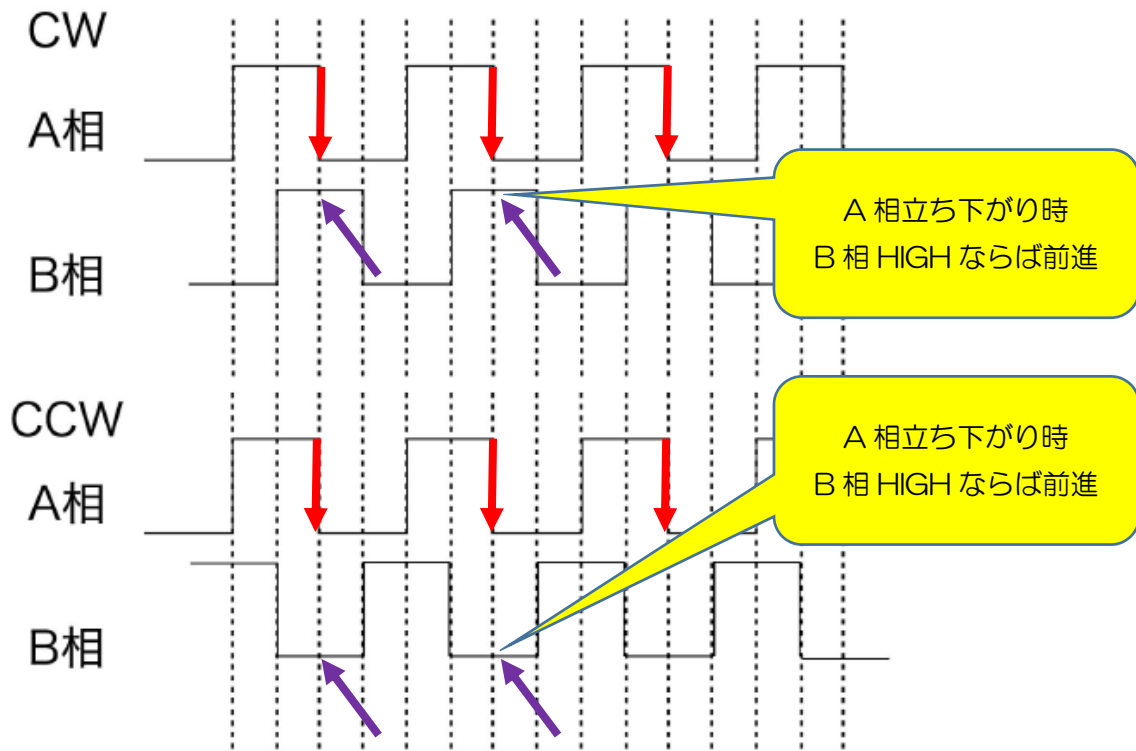
端子

モータ端子 M1 白 M2 赤
エンコーダ端子 黒 VDD
青 GND
緑 A 相
黄 B 相

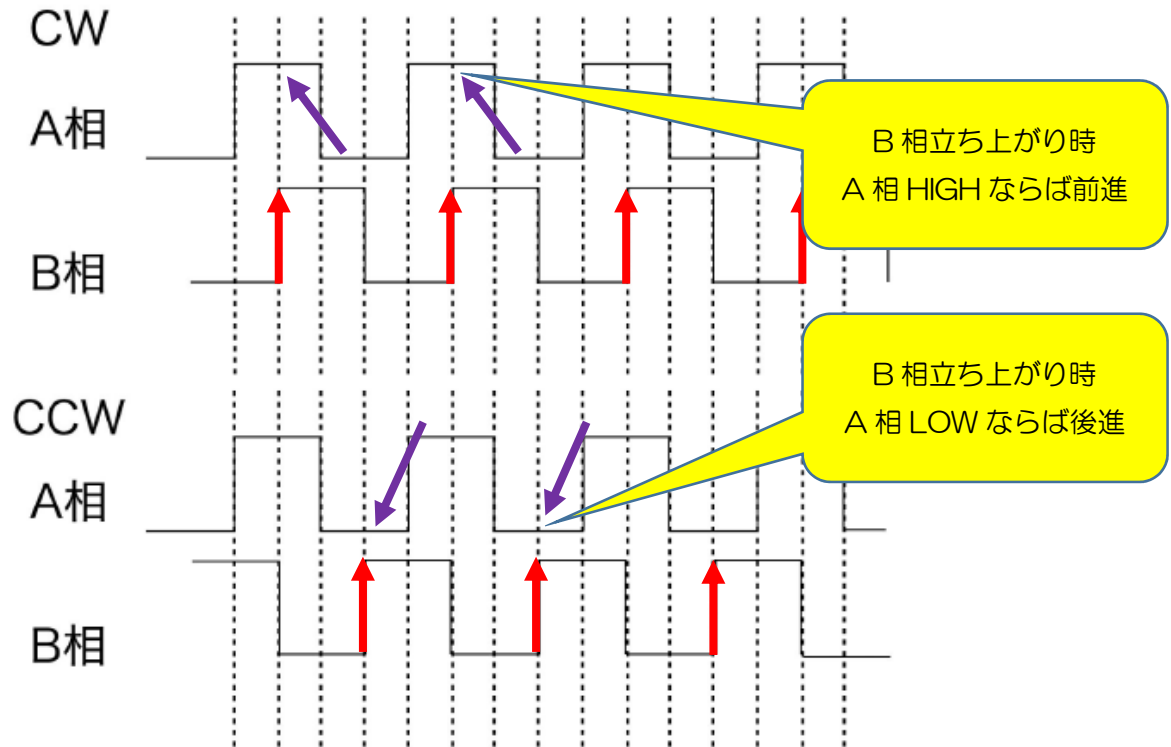
テキスト 53P. 動作確認

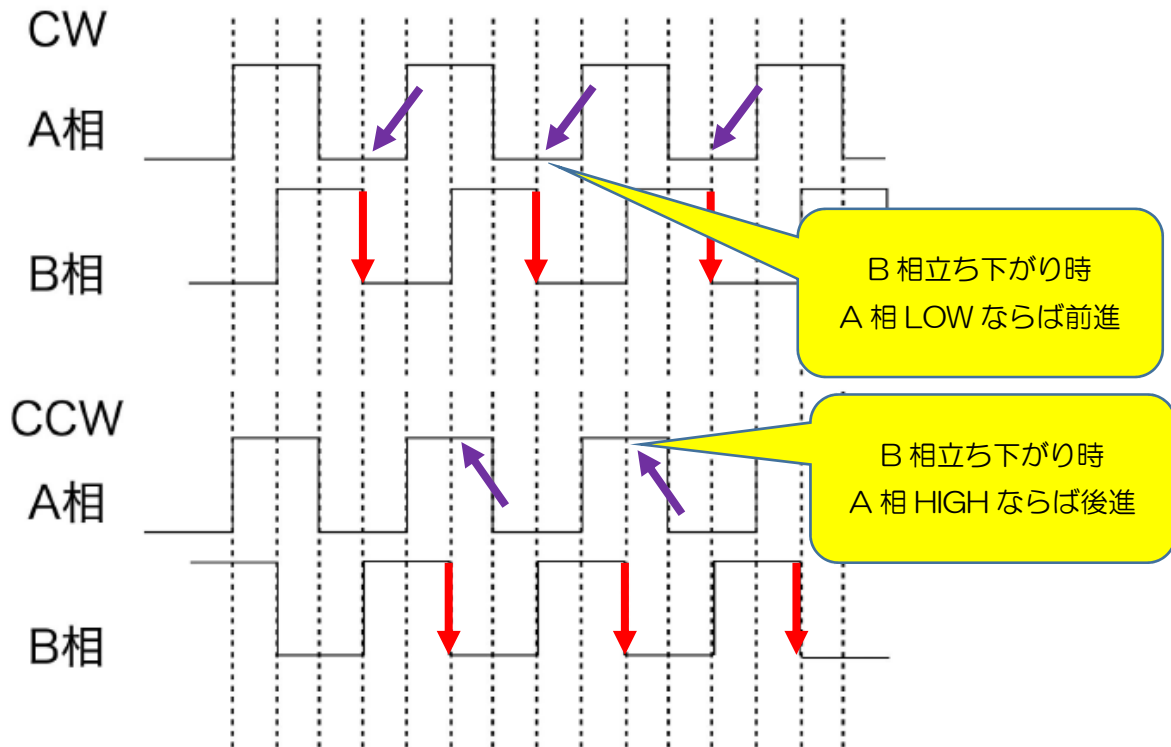
A 相を基準とした場合





B相基準とした場合





RT トレーニングトレーサ搭載エンコーダ

7ppr (pulse per revolution) (エンコーダ 1 回転で 7 パルス出力)

ギア比 1:19.35 だから出力軸 1 回転で $7 \times 19.35 = 135.5$ パルス

タイヤ径 $\phi 21\text{mm}$ タイヤ径 $\times \pi = 65.94\text{mm}$

タイヤ 1 回転 = 出力軸 1 回転 = $135.5\text{pulse} = 65.94\text{mm}$ の関係となる。

両エッジ(立ち上がり, 立ち下がり)でパルスカウントすると

$7 \times 4\text{ppr}$ (pulse per revolution) (エンコーダ 1 回転で $7 \times 4 = 28$ パルス出力)

ギア比 1:19.35 だから出力軸 1 回転で $28 \times 19.35 = 541.8$ パルス

タイヤ 1 回転 = 出力軸 1 回転 = $541.8\text{pulse} = 65.94\text{mm}$ の関係となる。

エンコーダにおいては、モータが回っている間、パルスがカウントアップまたは、カウントダウンしています。回転数は、rpm(レボリューション パー ミニッツ)という単位で表記することが多いです。rpmは1分間での回転数です。これも時間に関するため、一定の間隔で増加/減少したカウント値から算出する必要があります。例えば、100msの間に30パルス(4通倍)カウントアップされた場合は、

$$30/28/0.1 * 60 = 642[\text{rpm}]$$

ということになります。

RT トレーニングトレーサーを使った研修会(四,五日目)

エンコーダ値カウント関数は以下のようになる。

```
void EncoderA_R() {
    if( digitalRead(ENC_A_R_Pin)==HIGH ) {
        if( digitalRead(ENC_B_R_Pin)==LOW ) {
            Encoder_R_CNT++;
        } else {
            Encoder_R_CNT--;
        }
    } else {
        if( digitalRead(ENC_B_R_Pin)==HIGH ) {
            Encoder_R_CNT++;
        } else {
            Encoder_R_CNT--;
        }
    }
}

void EncoderB_R() {
    if( digitalRead(ENC_B_R_Pin)==HIGH ) {
        if( digitalRead(ENC_A_R_Pin)==HIGH ) {
            Encoder_R_CNT++;
        } else {
            Encoder_R_CNT--;
        }
    } else {
        if( digitalRead(ENC_A_R_Pin)==LOW ) {
            Encoder_R_CNT++;
        } else {
            Encoder_R_CNT--;
        }
    }
}

void EncoderA_L() {
    if( digitalRead(ENC_A_L_Pin)==HIGH ) { // 左 M の A 相が HIGH
        if( digitalRead(ENC_B_L_Pin)==LOW ) { // 左 M の B 相が LOW ならば
```

RT トレーニングトレーサーを使った研修会(四,五日目)

```

        Encoder_L_CNT++;           // 左 M 前進
    } else {                       // 左 M の B 相が HIGH ならば
        Encoder_L_CNT--;         // 左 M 後進
    }
} else {                           // 左 M の A 相が LOW
    if( digitalRead(ENC_B_L_Pin)==HIGH ) { // 左 M の B 相が HIGH ならば
        Encoder_L_CNT++;       // 左 M 前進
    } else {                     // 左 M の B 相が LOW ならば
        Encoder_L_CNT--;       // 左 M 後進
    }
}
}
}

void EncoderB_L( ) {
    if( digitalRead(ENC_B_L_Pin)==HIGH ) {
        if( digitalRead(ENC_A_L_Pin)==HIGH ) {
            Encoder_L_CNT++;
        } else {
            Encoder_L_CNT--;
        }
    } else {
        if( digitalRead(ENC_A_L_Pin)==LOW ) {
            Encoder_L_CNT++;
        } else {
            Encoder_L_CNT--;
        }
    }
}
}

```

カウントのための関数をどうやって動かすか。

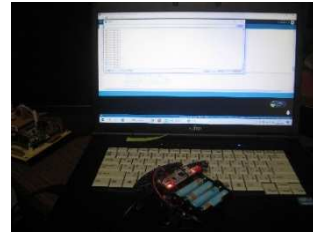
割り込み処理でエンコーダのカウントを行う。NUCLEO-F303K8 は割り込み端子として全ての端子が利用可能であり関数定義は

```
attachInterrupt( ピン番号, 割り込みで使用する関数名, エッジの種類);
```

となっている。

RT トレーニングトレーサーを使った研修会(四,五日目)

```
// attachInterrupt( ピン番号 , 割り込みで使用する関数名, エッジの種類 )
attachInterrupt( ENC_A_R_Pin, EncoderA_R, CHANGE );
attachInterrupt( ENC_B_R_Pin, EncoderB_R, CHANGE );
attachInterrupt( ENC_A_L_Pin, EncoderA_L, CHANGE );
attachInterrupt( ENC_B_L_Pin, EncoderB_L, CHANGE );
```



実習 1

トレーニングトレーサのタイヤを回転させ、画面にカウンタ値を表示させなさい。

```
int SW1_Pin = D7;
int SW2_Pin = D8;
:
int LED_Pin = D13;
int BUZZER_Pin = D2;
:
int ENC_A_R_Pin = A0;
int ENC_B_R_Pin = A1;
int ENC_A_L_Pin = D1;
int ENC_B_L_Pin = D9;
:
int Encoder_R_CNT = 0;
int Encoder_L_CNT = 0;
:
void EncoderA_R() {
:
}
void EncoderB_R() {
:
}
void EncoderA_L() {
:
}
void EncoderB_L() {
:
}
```

```
void setup() {
// put your setup code here, to run once:
:
pinMode(ENC_A_R_Pin, INPUT);
pinMode(ENC_B_R_Pin, INPUT);
pinMode(ENC_A_L_Pin, INPUT);
pinMode(ENC_B_L_Pin, INPUT);
:
attachInterrupt( ENC_A_R_Pin, EncoderA_R, CHANGE );
attachInterrupt( ENC_B_R_Pin, EncoderB_R, CHANGE );
attachInterrupt( ENC_A_L_Pin, EncoderA_L, CHANGE );
attachInterrupt( ENC_B_L_Pin, EncoderB_L, CHANGE );
:
Serial.begin(115200);
while(1) {
if( digitalRead(SW2_Pin)==LOW ) {
BUZZER_DRIVE(1, 100, 100);
while(1) {
Serial.printf(“%n¥r ENC_L=%d ENC_R=%d”,
Encoder_L_CNT, Encoder_R_CNT);
}
}
}
void loop() {
// put your main code here, to run repeatedly:
}
```

RT トレーニングトレーサーを使った研修会(四,五日目)

実習 2

トレーニングトレーサを 10cm 前進し、停止させなさい。

両エッジ(立ち上がり, 立ち下がり)でパルスカウントすると
 $7 \times 4 \text{ ppr}$ (pulse per revolution) (エンコーダ 1 回転で $7 \times 4 = 28$ パルス出力)
 ギア比 1:19.35 だから出力軸 1 回転で $28 \times 19.35 = 541.8$ パルス
 タイヤ径 $\phi 21 \text{ mm}$ タイヤ径 $\times \pi = 65.94 \text{ mm}$
 タイヤ 1 回転=出力軸 1 回転= $541.8 \text{ pulse} = 65.94 \text{ mm}$ の関係となる。

10cm=100mm 進むためには

$$541.8 \times 100 \div 65.94 = 821.6 \div 822 \text{ pulse}$$

10cm 走行=822pulse 走行となる。

```
Encoder_C_CNT = 0;
while( Encoder_C_CNT < 822 )
{
    :
    Encoder_C_CNT = (Encoder_L_CNT + Encoder_R_CNT)/2 ;
}
run_stop();
```

2.コースを記憶する

ロボトレース競技のコース長は最大 60m と設定されています。
 カーブの最小局半径 10cm ということもあり、10cm を 1 区画とし
 コース全長を覚えることとする。

(1) 区画数

$$60 \text{ m} \div 10 \text{ cm} = 600 \text{ 区画}$$

(2) 確保

```
int wmap[ 600 ], wmap2[600] wmap3[600] 確保
```

(3) wmap[] の内容

初期値 0、ライントレースの制御量(Line_Control)を記録

(4) wmap2[]の内容

wmap[]から作成。初期化 0、

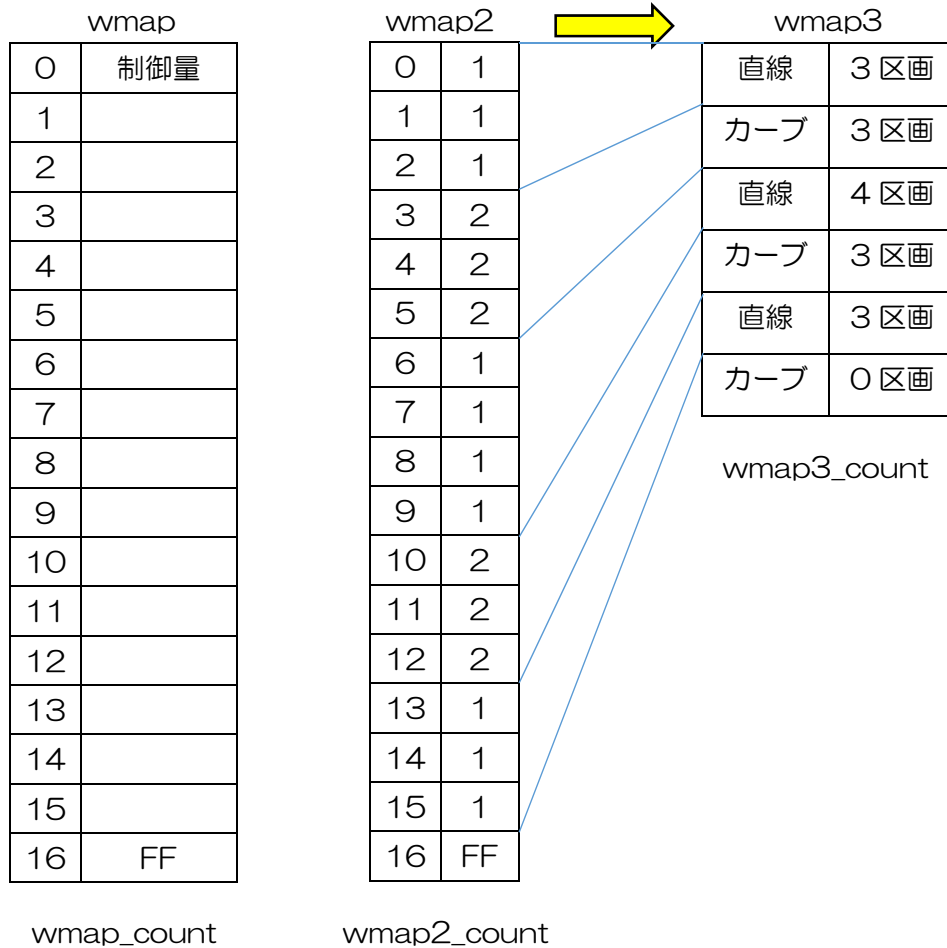
その区画が直線ならば 1、カーブなら 2 を記録。

RT トレーニングトレーサーを使った研修会(四,五日目)

(5) 判断

直線, カーブ区画の判断は 制御量の平均等を考えたいが、計算量が多くなるので、今回は Line_Control の絶対値が 300 より小さい場合は直線区画、大きい場合はカーブ区画と判断する。(※この判断は今後詳細を詰める必要あり)

(6) 例



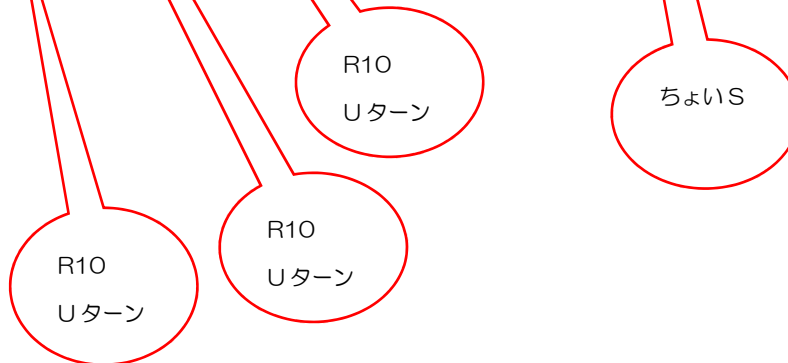
実習 3

練習コースを走行し、コースデータを表示させなさい。

RT トレーニングトレーサーを使った研修会(四,五日目)

1	wmap[0]= 368	wmap2[0]= 2	34	wmap[33]= -152	wmap2[33]= 1
2	wmap[1]= 104	wmap2[1]= 1	35	wmap[34]= 43	wmap2[34]= 1
3	wmap[2]= -40	wmap2[2]= 1	36	wmap[35]= -135	wmap2[35]= 1
4	wmap[3]= 43	wmap2[3]= 1	37	wmap[36]= -168	wmap2[36]= 1
5	wmap[4]= 206	wmap2[4]= 1	38	wmap[37]= -203	wmap2[37]= 1
6	wmap[5]= -101	wmap2[5]= 1	39	wmap[38]= 75	wmap2[38]= 1
7	wmap[6]= -16	wmap2[6]= 1	40	wmap[39]= 175	wmap2[39]= 1
8	wmap[7]= 46	wmap2[7]= 1	41	wmap[40]= -45	wmap2[40]= 1
9	wmap[8]= 28	wmap2[8]= 1	42	wmap[41]= 36	wmap2[41]= 1
10	wmap[9]= 265	wmap2[9]= 1	43	wmap[42]= 70	wmap2[42]= 1
11	wmap[10]= 85	wmap2[10]= 1	44	wmap[43]= 43	wmap2[43]= 1
12	wmap[11]= 163	wmap2[11]= 1	45	wmap[44]= 52	wmap2[44]= 1
13	wmap[12]= 29	wmap2[12]= 1	46	wmap[45]= 205	wmap2[45]= 1
14	wmap[13]= 261	wmap2[13]= 1	47	wmap[46]= 202	wmap2[46]= 1
15	wmap[14]= 142	wmap2[14]= 1	48	wmap[47]= -399	wmap2[47]= 2
16	wmap[15]= -69	wmap2[15]= 1	49	wmap[48]= 25	wmap2[48]= 1
17	wmap[16]= -653	wmap2[16]= 2	50	wmap[49]= -218	wmap2[49]= 1
18	wmap[17]= -1247	wmap2[17]= 2	51	wmap[50]= 551	wmap2[50]= 2
19	wmap[18]= -777	wmap2[18]= 2	52	wmap[51]= 614	wmap2[51]= 2
20	wmap[19]= -167	wmap2[19]= 1	53	wmap[52]= 292	wmap2[52]= 1
21	wmap[20]= 193	wmap2[20]= 1	54	wmap[53]= 188	wmap2[53]= 1
22	wmap[21]= 998	wmap2[21]= 2	55	wmap[54]= 267	wmap2[54]= 1
23	wmap[22]= 1060	wmap2[22]= 2	56	wmap[55]= 42	wmap2[55]= 1
24	wmap[23]= 114	wmap2[23]= 1	57	wmap[56]= -186	wmap2[56]= 1
25	wmap[24]= 232	wmap2[24]= 1	58	wmap[57]= -110	wmap2[57]= 1
26	wmap[25]= 319	wmap2[25]= 2	59	wmap[58]= -471	wmap2[58]= 2
27	wmap[26]= 261	wmap2[26]= 1	60	wmap[59]= -373	wmap2[59]= 2
28	wmap[27]= -140	wmap2[27]= 1	61	wmap[60]= -553	wmap2[60]= 2
29	wmap[28]= -727	wmap2[28]= 2	62	wmap[61]= -251	wmap2[61]= 1
30	wmap[29]= -619	wmap2[29]= 2	63	wmap[62]= -34	wmap2[62]= 1
31	wmap[30]= -425	wmap2[30]= 2	64	wmap[63]= -175	wmap2[63]= 1
32	wmap[31]= -336	wmap2[31]= 2	65	wmap[64]= -172	wmap2[64]= 1
33	wmap[32]= -231	wmap2[32]= 1	66	wmap[65]= -77	wmap2[65]= 1

データを見ての考察



データを見て練習コース難所も判断することができる。

これで第 2 走行の準備は整った。

このデータを利用し各自第 2 走行のプログラム工夫して下さい。

本日はここまで。第六日目に続く